Software Inspections and the Year 2000 Problem

Don O'Neill
Independent Consultant

The year 2000 (Y2K) problem promises to impact information systems of all kinds. With no silver bullet available, the responsible manager must take a variety of actions to detect and correct this problem. This article provides a compliance checklist to aid in Y2K problem detection and correction.

Software inspections are a powerful mechanism to help detect and correct the Y2K problem. Software inspections call for a close and strict examination of software artifacts against the standard of excellence set by the organization, significantly improving defect detection and removal capabilities. By explicitly setting the standard of excellence for software products to operate correctly before, during, and after 2000, and by conducting software inspections on software products thought to be date sensitive, the responsible manager is taking an important and prudent step to meet the challenge.

To reason effectively about Y2K compliance at the program level, the reviewer must understand the standard date format and treatment by system date routines, identify and verify internal date usage, identify and verify external file date usage, and understand the context of date usage within the application domain. Specifically:

- Reasoning about internal date usage includes verifying date representations, evaluating date transformations and logical expressions, identifying variable names that contain date types, and recognizing and handling the leap year anomaly. Not all problems lie in looking forward; looking-backward calculations must also be identified and assessed.
- Reasoning about external file date usage includes identifying external system interfaces and verifying standard date format for input and output date records.
- Reasoning about the context of date usage within the application domain requires an understanding of the time horizon to failure (THF). For example, the THF for a 30-year mortgage is 1970; for an enterprise five-year plan, 1995; for a four-year motor vehicle license, 1996; for a two-year credit card issuance, 1998; and for various annual deadlines, 1999.

To help practitioners uncover all possible Y2K problem situations, the following Y2K compliance checklist is added to the standard of excellence set by the organization. This checklist is drawn from the software inspections course and lab I offer. These checklists are organized along a common framework that includes completeness, cor-

Year 2000 Compliance Checklist

1.	Has the product component been assessed for Y2K compliance?
1.1	Is Y2K compliance specified as standard date format (such as, YYYYMMDD)?
1.2	Are date-related system services identified and Y2K compliant?
1.2.1	Are system date routines identified and Y2K compliant?
1.3	Are all date-related nodes and flow graphs identified and Y2K compliant?
1.3.1	Is internal date usage identified and Y2K compliant?
1.3.1.1	Are all date representations identified and Y2K compliant?
1.3.1.2	Are all date transformation routines identified and Y2K compliant?
1.3.1.3	Are all date-related names identified and Y2K compliant?
1.3.1.4	Are all date calculations identified and Y2K compliant?
1.3.1.5	Are all date uses in logical expressions identified and Y2K compliant?
1.3.1.6	Are all leap year computations identified and Y2K compliant?
1.3.1.7	Are all "looking backward" (from the year 2000) calculations identified and assessed as Y2K compliant?
1.3.1.8	Are all "looking forward" (from the year 2000) calculations identified and assessed as Y2K compliant?
1.3.2	Are all files identified and Y2K compliant?
1.3.2.1	Are all external system interfaces identified and Y2K compliant?
1.3.2.2	Are all input and output date records formatted as standard date format (such as, YYYYMMDD) and Y2K compliant?
1.3.2.3	Are all extended semantics (embedded dates and sort keys) identified and Y2K compliant?
1.3.2.4	Are all imported files identified and Y2K compliant?
1.3.3	Has the Year 2000 Time Horizon to Failure (THF) been identified for the application?
1.3.3.1	What is the THF for the application?

rectness, style, rules of construction, multiple views, technology, and metrics. The Year 2000 Compliance Checklist is one of the multiple views for design and code. For more information on this software inspections training and the results it obtains, please visit http://members.aol.com/ONeillDon/index.html. ◆

About the Author

Don O'Neill is an experienced software engineering manager and technologist currently serving as an independent consultant. Following 27 years with IBM's Federal Systems Division, he completed a three-year residency at Carnegie Mellon University's Software Engineering Institute under IBM's Technical Academic Career Program. As an independent consultant, O'Neill conducts defined programs to manage strategic software improvement, including an organizational software inspections process, directing the National Software Quality Experiment, implementing software risk management on the project, conducting the Project Suite Key Process Area Defined Program, and conducting Global Software Competitiveness assessments. He served on the executive board of the Institute of Electrical and Electronics Engineers (IEEE) Software Engineering Technical Committee and as a distinguished visitor of the IEEE. He is a founding member of the National Software Council and the Washington, D.C. Software Process Improvement Network.

9305 Kobe Way
Gaithersburg, MD 20879
Voice: 301-990-0377
Fax: 301-670-0234
E-mail: ONeillDon@aol.com
Internet: http://members.aol.com/ONeillDon/index.html

Coming Events

Ownership and Control Issues in Architecture-Based Acquisition of Product Lines

Dates: Jan. 13-14, 1998
Location: Pittsburgh, Pa.
Subject: The objective of this meeting is to produce a short point paper articulating the acquisition business models identified at the Salem '97 workshop.
Sponsor: Software Engineering Institute
Contact: James Withey
E-mail: jvw@sei.cmu.eu

Configuration Management Seminars: (1) Basic, (2) Advanced, (3) Comprehensive

Locations: Bethesda, Md.; San Diego, Calif.; Washington, D.C.; Las Vegas, Nev.
Dates: Jan. 26 – March 11. Contact Dana Marcus for specific date of each seminar.
Subjects: (1) Basic configuration management (CM) course for individuals in the configuration field for six months or longer. The latest CM standards and requirements; scope and elements of a good CM plan; English release requirements; establishing appropriate baselines; managing CM status accounting records and reports; guidelines for handling and documenting variances, etc. (2) Advanced CM course for individuals possessing the basic knowledge of the CM field; impact of COTS, NOTS, and NDI on CM requirements for Department of Defense procurement; developing models and metrics for CM products and processes; establishing comprehensive change management and corrective action systems. (3) A comprehensive methodology for implementing CM; incorporating the best practices from industry leaders, the latest technology, and the newest standards, guidelines, and requirements, including the new standards J-STD-016, ISO122207, US 122207, EIA-649, and MIL-HNBK-61.
Sponsor: Technology Training Corporation
Contact: Dana Marcus
Voice: 310-534-3922
E-mail: dmarcus@ttcus.com

First Workshop on Biologically Inspired Solutions to Parallel Processing Problems (BioSP3)

Dates: March 30 – April 3, 1998
Location: Orlando, Fla.
Subject: This workshop seeks to provide an opportunity for researchers to explore the connection between biologically based techniques and the development of solutions to problems that arise in parallel processing.
Sponsor: IEEE Technical Committee on Parallel Processing (tentative)
Contact: http://www.ee.uwa.edu.au/staff/zomaya.a.html/BioSP3.html

IEEE Computer Society International Conference on Computer Languages 1998

Dates: May 14-16, 1998
Location: Loyola University Chicago, Chicago, Ill.
Sponsor: IEEE Computer Society, Technical Committee on Computer Languages in cooperation with the Association for Computing Machinery Special Interest Group on Programming Languages.
Contact: http://www.math.luc.edu/iccl98/